IN THE SPECIFICATION:

Please substitute the following paragraphs with the paragraphs originally submitted in the application:

On page 1, please substitute the paragraph beginning on line 24 and ending on line 30 with:

Several methods ~~are so far~~ have previously been proposed as conventional techniques. US. Paent No. 5,850,592, entitled "Method for self-organizing wireless station network", discloses a group of wireless devices that automatically organizes or configures itself into a multi-layered network for relaying messages from station to station, with some of the stations operating at a relatively high power level as message gateways for a cluster of stations operating at lower power, thereby forming a network backbone providing longer distance network links through the gateways.

On page 2, please substitute the paragraph beginning on line 33 and ending on line 35 with:

Whereas the above conventional arts ~~has~~ have been proposed, there are continuous needs to provide a method, a system and a program product for efficiently organizing devices in a wireless transmission ~~system~~ system into bounded size clusters in a short amount of time.

On page 3, please substitute the paragraph beginning on line 3 and ending on line 16 with:

The present invention has been made essentially based on the following ideas[[;]]:

a) obtaining some information about the state of the device in the device discovery messages is more time efficient than setting up a connection with the devices to obtain such information, and

that

b) In the case when the devices need to discover the identity of other devices in vicinity, the process to form a wireless network will take less time when one set of these devices has the job of only transmitting the inquiry message and the rest of the devices have the job of only listening out for any such messages and sending responses to them. Also, the entire clustering process will take lesser time when the devices carry out the task of transmitting inquiry messages and the task of listening for these inquiry messages continuously instead of intermittently. We will refer to the devices transmitting the inquiry messages as devices being in the transmit-state and the devices listening out for these messages as ~~devies~~ devices being in the receive-state. In Bluetooth these states correspond to Inquiry and Inquiry-scan states respectively.

On page 4, please substitute the paragraph beginning on line 35 and ending on line 35 with:

Fig. 7(a), (b) shows the flowchart of a ~~proeeess~~ process executed for Super-Master-election.

On page 5, please substitute the paragraph beginning on line 24 and ending on line 32 with:

The second technique is achieved by setting the value of the Inquiry-scan interval (the period of time between the start of two successive Inquiry-scan periods, referred to as Inquiry-scan-interval in the Bluetooth Standard) ~~(http://www.bluetooth.net/)~~ close to or equal to the value of the Inquiry-scan window. The closer the value of the scan interval becomes to the value of the scan window, the clustering operation will become faster. For achieving interconnection between piconets, the separation of the inquirers and scanners may be achieved by the delegation

of the inquiry and scan procedures to specific Slaves in the piconet. The Master of the piconet may keep the other existing connections from getting uninterrupted.

On pages 9-10, please substitute the paragraph beginning on page 9, line 26 and ending on page10, line 11 with:

Fig. 1 shows a flowchart of the algorithm executed by a Master-designate. If the node is designated as a Master-designate in the step 101, the Master designate sets state=INQUIRY, ~~number_of_responces=0~~ number_of_responses=0, number_of_masters=0, and response_list=null in step 102. In step 103, the algorithm determines whether the Master-designate receives the CLUSTER_TO or ~~SUPER_TO~~ SUPERM_TO. If the CLUSTER_TO or ~~SUPER_TO~~ SUPERM_TO is not reached (yes) the algorithm starts the Master-designate to broadcast Inquiry packet including the bit carrying the specific information in the step 104. The algorithm proceeds to the step 105 to determine whether or not any response is reached from the Slave designate. When the response is received from a Slave-designate (yes) in the step 105, the algorithm sets ~~number_of_responces~~ number_of_responses to ++, and adds ID and Clock of sender to response_list in the step 106 and the Master-designate makes the Slave in its cluster by paging and making a connection to it as long as the maximum cluster size is not exceeded. The algorithm next determines in the step 107 if this Slave is the first Slave of the cluster (number_of_responses==1). If so (yes), the Master-designate set the Slave to be a Proxy-slave in the step 108. When this Slave is not the first one (no), then the algorithm proceeds to the step 109 and determines the cluster becomes full. When the cluster becomes full (yes), the Master-designate sets itself as Master in the step 110, and any future inquiry responses from Slave-

designates are ignored. If not (no), the algorithm next proceeds to the step 111 and determines whether or not the response from the Proxy-Slave is received. When the response from the Proxy-slave is received (yes), the algorithm sets number_of_masters to be ++ in the step 112.

On pages 10, please substitute the paragraph beginning on line 12 and ending on line 29 with:

As part of the Super-master election which is interleaved with the cluster formation from the step 113 to the step 117, the algorithm proceeds to the step 113 and determines whether or not CLUSTER_TO occurs; ~~numebr_of_masters~~ number_of_masters = =k or SUPERM_TO occurs and node has become the Master. If so (yes), the algorithm proceeds to the step 114 and the Master-designate or Master is set as the Super-master-designate and the information is transmitted to the Proxy-Slave. If not (no), the algorithm further proceeds to the step 115 and determines whether the node has not become Master-designate by stage 1 and CLUSTER_TO occurs. If so (yes), the algorithm proceeds to the step 116 to end. If not (no), the algorithm further determines whether or not CLUSTER_TO occurs and number_of_response is equal to 0 in the step 117 and if not (no) the algorithm goes back the while -loop started from 103. If so (yes), the algorithm proceeds to the step 118 and set node as Slave-designate and reverts to the step 103. Namely, in the algorithm from the step 113 to 118, the Master/Master-designate collects up to *k* responses from Proxy-slaves of other clusters, or times out (SUPERM_TO). At this point, the node declares itself Super-master-designate. However, this happens only after CLUSTER_TO has occurred. If the Master-designate has not collected any responses by the CLUSTER_TO period, then it becomes a Slave-designate and starts scanning. A sample pseudo-code for executing the Master-designate from the step 101 to 118 is shown in Fig. 2.

On pages 10-11, please substitute the paragraph beginning on page 10, line 30 and ending on page 11, line 18 with:

Fig. 3 shows a flowchart of the algorithm executed in the Slave-designate. The algorithm shown in Fig. 3 first determines whether the node is Slave-designate in the step 301. The algorithm then proceeds to the step 302 to set the state =INQUIRY_SCAN and starts the Slave-designate continuously to scan for Inquiry messages. The algorithm proceeds to the while-loop and determines whether or not the node is a ~~Slave-destinate~~ Slave-designate in the step 303, and if so (yes) the algorithm then proceeds to the step 304 and determines whether Inquiry packets are received. If so (yes), the Slave-designate sends Inquiry_response packet and set state=PAGE_SCAN in the step 305. The algorithm next determines whether or not Page packet is received immediately in the step 306. If Inquiry packet is received immediately (yes), then the algorithm proceeds to the step 307 to complete connection establishment; to set state = CONNECTED and set the node to be Slave. If not (no), the algorithm then proceeds to the step 308 and set state =Inquiry_SCAN to revert the step 303. The algorithm then proceeds to the step 308 to determine whether or not Master asks the Slave to become Proxy-Slave. If so (yes), the algorithm proceeds to the step 309 to set the node to be Proxy-Slave. If not (no), the algorithm proceeds to the step 310 and the Proxy-Slave is not paged and the state is set to be INQUIRY_SCAN and goes back to the step 303. These procedures are executed whenever the node is a Slave-designate as shown in the steps 303~310. Fig, 4 shows a sample pseudo-code for executing the procedure shown in Fig. 3. Summarizing the procedure shown in Fig. 3, if on sending an inquiry response, the inquirer does not page it and does not establish a connection, then it goes back to the scan state. However, if the inquirer connects to it, then it becomes a

A9 cont

Slave of the Master, i.e., clusterheaded by its inquirer, and steps scanning. If the Master/Master-designate directs it to become a Proxy-slave, it goes into scan for the Super-master election.

On page 12, please substitute the paragraph beginning on line 21 and ending on line 32 with:

A10

Fig. 7 shows a flowchart of the algorithm executed by the Super-master-designate. The algorithm executed by the Super-master-designate first determines that the node is a Super-master-designate in the step 701 and the algorithm proceeds to the while-loop started from the step 702. If the node is a Super-master-designate (yes), the algorithm proceeds to the step 703 and collect cluster information from all of the Proxy-Slaves which responded. In the step 702, in order to make a connection to the Proxy-Slave, the Super-master-designate transmit the Inquiry packet so that the algorithm jumps to PAGE_SCAN soon after responding and the connection has been established. The algorithm next proceeds to the step 704 and determines whether or not number_of_masters is less than k. If so (yes), since ~~SUPER_TO~~ SUPERM_TO has occurred, then the algorithm proceeds to the step 705 and makes a connection to the Proxy-slave having the highest ID and sets the node to be Super-master and the algorithm reverts to the step 713.

On page 13, please substitute the paragraphs beginning on line 2 and ending on line 34 with:

A8

If not (no), the algorithm shown in Fig. 7 (a) then proceeds to the step 708 and determines whether or not the total number of known nodes in k clusters is less than N. If yes (so), the algorithm shown in Fig. 7 (a) further proceeds to the step 709 and determines whether or not new Proxy-Slaves respond to Inquiry packets. If so (yes), the ~~algorithmfurther~~ algorithm further proceeds to the step 710 and sets numbers_of_masters to be ++ and collects clusters information from the new Proxy-Slaves. If not (no), the algorithm further proceeds to the step 711 and

determines whether the total number of known nodes = =N. If so (yes), the algorithm proceeds to the step 712 and the nodes set the highest ID Master to be Super-master and also instructs the Super-master to terminate and gives the Super-master all X Master IDs and the algorithm reverts to the step 713. The algorithm shown in Fig. 7 (a) exit the loop started from the step 702 until the mode is not the Super-master-designate.

The algorithm executed by the ~~Super-master-deisgnate~~ Super-master-designate following to Fig. 7 (a) proceeds to Fig. 7 (b) and the algorithm further determines whether the node has been asked by the Proxy-Slave to be a Super-master and all N nodes are not known in the step 713. If so (yes), the algorithm further determines whether or not all of the N nodes is not known in the step 714. If so (yes), further the algorithm determines whether (k-X) is less than or equal to or less than S in the step 715 and if so (yes), the algorithm makes all (k-X) slaves to be new Masters in the step 716 which then collect the remaining nodes. If not (no), the algorithm makes S slaves new Masters which collect new Slave-designates in the step 717; the first (k-X-S) Slave-designates replying to it are made to be Masters. The algorithm further proceeds to the step 718 and determines whether or not the new Slave-designates respond to Inquiry packets. If so (yes), these are orphan Slave-designates which are not part of any cluster, and the algorithm further determines whether or not the node's cluster has numbers of Slaves less than S in the step 719. If so (yes), the algorithm ~~add~~ adds a new Slave to the cluster as the Master-designate does in the step 720, end if not (no), the algorithm ~~add~~ adds to a separate list of additional slaves in the step 721 and the algorithm goes back to the loop started from the step 714. If the determination in the step 714 is no, the algorithm further determines whether the number_of_clusters is larger than k in the step 801 shown in Fig. 8, and if so (yes), the algorithm tears down excess clusters and

distribute the nodes among k clusters; connects the clusters to form a desired topology by deciding the bridges and neighboring clusters for every Masters; and informing the nodes in the step 802. The algorithm further proceeds to the step 803 and transmits a termination message to all clusters (and hence all nodes). A sample pseudo-code is shown in Fig. 9.